AMENDMENTS TO THE CLAIMS:

1-20 (Cancelled)

21. (Previously Presented) A virtualization switch for performing a plurality of virtualization services within a data path said virtualization switch comprises:

a network interface (NI);

an iSCSI module;

a target manager (TM);

a volume manager (VM) capable of translating a logic command to a list of physical commands, wherein said physical commands are constructed in a data structure, said data structure defines the relations between said physical commands;

a data transfer arbiter (DTA);

a device manager (DM);

a plurality of input ports to receive incoming packets from a network; and,

a plurality of output ports to communicate with plurality of storage devices.

22. (Original) The virtualization switch of claim 21, wherein said data structure comprises at least one of: alternative command link, pointer to said storage device.

23. (Original) The virtualization switch of claim 22, wherein said alternative command link links between at least two physical commands that can be executed in parallel.

24. (Original) The virtualization switch of claim 21, wherein said VM further comprises a mapping schema uses for translating said logic command to said list of said physical commands.

25. (Original) The virtualization switch of claim 24, wherein said mapping schema defines relations between virtual volumes, logical units (LUs), and said storage devices.

26. (Original) The virtualization switch of claim 25, wherein said virtual volume is at least one of: concatenation volume, stripe volume, mirrored volume, simple volume, snapshot volume.

27.-37 (Cancelled)

38. (Previously Presented) A method for performing a plurality virtualization services, said method being further operative to perform said virtualization services within a data path, said method comprises the steps of:

a) receiving a logic command to be performed on at least one virtual volume, said logic command including at least a virtual address;

d) scheduling said logic command for execution, wherein said logic command is at least a SCSI command comprising the steps of:

1) initiating an iSCSI session with an initiator host including steps of

a) determining if said initiator host is authorized to send said logic command; and,

b) denying said logic command from said initiator host, if said initiator host is unauthorized;

2) receiving said logic command from said initiator host;

3) parsing said logic command to determine at least said virtual address and said logic command's type;

4) performing a check to determine if said logic command is valid;

5) generating a response command if said logic command is invalid, otherwise, adding said logic command to a host-LU queue; and,

6) generating a data transfer request;

c) translating, in one pass, said logic command to a list of physical commands, wherein each of said physical commands is targeted to a different storage device;

d) determining the amount of data to be transferred via a network; and,

e) executing said physical commands on said storage devices.

39. (Previously Presented) The method of claim 38, wherein said response command comprises an iSCSI service response code indicating the type of a generated error.

40. (Previously Presented) The method of claim 38, wherein said host-LU queue comprises logic commands requested to be executed by said host on said LU.

41. (Previously Presented) The method of claim 38, wherein scheduling said logic command for execution further comprises the step of: selecting said logic command to be executed from said host-LU queue.

42. (Original) The method of claim 41, wherein the selection is performed using at least one of the following selection algorithms: recently used, round robin, weighted round robin, random, least loaded LU.

43. (Previously Presented) The method of claim 38, wherein said command type is a read command.

44. (Original) The method of claim 43, wherein said amount of data to be transferred is determined by an available space parameter.

45. (Original) The method of claim 44, wherein said available space parameter defines the number of data bytes to be sent to the host.

46. (Original) The method of claim 44, wherein the following steps comprise executing said physical commands on said storage devices:

a) accessing a storage device using a physical address;

b) retrieving from said accessed storage device the number of bytes designated in said available space parameter;

c) sending the retrieved data to said host; and,

d) repeating said steps a) through d) until all requested data is read from said storage devices.

47. (Original) The method of claim 46, wherein said physical commands are executed in parallel.

48. (Previously Presented) The method of claim 38, wherein said command type is a write command.

49. (cancelled)

50. (Currently Amended) A method for performing a plurality virtualization services, said method being further operative to perform said virtualization services within a data path, said method comprises the steps of:

a) receiving a logic command to be performed on at least one virtual volume, said logic command including at least a virtual address;

~~d)~~b) scheduling said logic command for execution;

c) translating, in one pass, said logic command to a list of physical commands, wherein each of said physical commands is targeted to a different storage device;

d) determining using a check point list the amount of data to be transferred via a network; wherein said check-point list further defines how data should be sent from an initiator host to said storage devices and;

e) executing said physical commands on said storage devices .

51. (Original) The method of claim 50, wherein said check-point list comprises a linked list of data chunks.

52. (Original) The method of claim 51, wherein the following steps comprise executing said physical commands on said storage devices:

a) filling at least one data chunk with said data retrieved from the network;

b) accessing said storage device using a physical address;

c) writing said data chunk to said accessed storage device; and,

d) repeating said steps a) through d) for all data chunks in said check-point list.

53. (Original) The method of claim 52, wherein said physical commands are executed in parallel.

54. (Previously Presented) The method of claim 53, wherein said physical commands are constructed in a data structure.

55. (Original) The method of claim 54, wherein said data structure further includes at least one of: an alternative command link, a pointer to said storage device.

56. (Original) The method of claim 55, wherein said alternative command link links between at least two physical commands that can be executed in parallel.

57.-67 (Cancelled)

68. (Previously Presented) A computer executable code for performing a plurality virtualization services stored on a recordable media, said computer executable code being further operative to perform said virtualization services within a data path, said code comprises the steps of:

a) receiving a logic command to be performed on at least one virtual volume, said logic command including at least a virtual address, wherein said logic command is at least a SCSI command comprising the steps of:

1) initiating an iSCSI session with an initiator host including steps of

a) determining if said initiator host is authorized to send said logic command; and,

b) denying said logic command from said initiator host, if said initiator host is unauthorized;

2) receiving said logic command from said initiator host;

3) parsing said logic command to determine at least said virtual address and said logic command's type;

4) performing a check to determine if said logic command is valid;

5) generating a response command if said logic command is invalid, otherwise, adding said logic command to a host-LU queue; and,

6) generating a data transfer request;

b) scheduling said logic command for execution;

c) translating, in one pass, said logic command to a list of physical commands, wherein each of said physical commands is targeted to a different storage device;

d) determining the amount of data to be transferred via a network; and,

e) executing said physical commands on said storage devices.

69-79 (Cancelled)

80. (Currently Amended) A computer product stored on a computer-readable medium comprising software instructions operable to enable a computer to perform a process for performing a plurality virtualization services, said process being further operative to perform said virtualization services within a data path, said code comprises the steps of:

a) receiving a logic command to be performed on at least one virtual volume, said logic command including at least a virtual address;

~~d)~~b) scheduling said logic command for execution;

c) translating, in one pass, said logic command to a list of physical commands, wherein each of said physical commands is targeted to a different storage device;

d) determining, using a check-point list, the amount of data to be transferred via a network wherein said a check-point list further defines how data should be sent from an initiator host to said storage devices; and,

e) executing said physical commands on said storage devices.

81. (Previously Presented) The computer program product of claim 80, wherein said check-point list comprises a linked list of data chunks.

82. (Previously Presented) The computer program product of claim 81, wherein the following steps comprise executing said physical commands on said storage devices:

a) filling at least one data chunk with said data retrieved from the network;

b) accessing said storage device using a physical address;

c) writing said data chunk to said accessed storage device; and,

d) repeating said steps a) through d) for all data chunks in said check-point list.

83. (Previously Presented) The computer program product of claim 82, wherein said physical commands are executed in parallel.

84. (Previously Presented) The computer program product of claim 83, wherein said physical commands are constructed in a data structure.

85. (Previously Presented) The computer program product of claim 84, wherein said data structure further includes at least one of: an alternative command link, a pointer to said storage device.

86. (Previously Presented) The computer program product of claim 85, wherein said alternative command link links between at least two physical commands that can be executed in parallel.

87-91 (Cancelled)